浙江水利水电学院

课程设计指导书

（ 2019/2020 学年 2 学期）

|  |  |
| --- | --- |
| 课程设计  名 称： | 数据结构与算法课程设计 |
| 教学单位： | 信息工程与艺术设计学院 |
| 专 业： | 软件工程 |
| 班 级： | 软件工程18-12 |
| 指导教师： | 寿焕君；张颖 |

2020年6月

# 一、目的和意义

《数据结构与算法课程设计》是课程教学中的一个重要的实践性环节，是在完成课堂教学的基础上进行的一项综合性训练。通过两周的课程设计，可以使学生加深对各种数据结构的逻辑特征、存储方式和基本操作的理解，加强分析数据、组织数据的能力，进一步掌握一些常用算法的实现过程，从而把书本知识运用到解决实际问题之中去，培养从事计算机工作所需要的动手能力。同时，也能把书上学到的知识变“活”，起到深化理解和灵活掌握教学内容的目的。在面对实际应用问题时，会选择合适的数据结构和存储结构，设计高效的算法，编制出正确且有效的程序。

# 二、基本要求

按“设计内容”中的要求，两个学生一组完成各个题目，学生要发挥自主学习的能力，合理安排好时间，制订课程设计进度表，并在课程设计过程中不断检测自己的计划完成情况，及时地向教师汇报。

本次课程设计以**C语言**为编程语言，各函数功能的划分要合理，数据的输入、输出要有提示信息。同学们要发挥各自的主动性，设计出高效的算法。

函数、变量的取名要符合一定规则，做到“见名知义”，不要重名。注意随时做好备份，以防数据丢失。

每组同学建立“**学号姓名学号姓名**”文件夹，内设五个子文件夹，分别以设计题目命名，如“舞会配对问题”。所有设计成果请及时备份，并妥善保管。

上机时要遵守机房的规章制度，爱护实验设备。

# 三、设计内容

## 1.采花生问题

宠物猴小Q发现路边有一块花生田，花生植株整齐地排列成矩形网格（如图1）。有经验的小Q一眼就能看出，每棵花生植株下的花生有多有少。为了训练小Q的算术，现规定：先找出花生最多的植株，去采摘它的花生；然后再找出剩下的植株里花生最多的，去采摘它的花生；依此类推，不过一定要在**限定的时间**内回到路边。（文件取名为**peanut.c**）

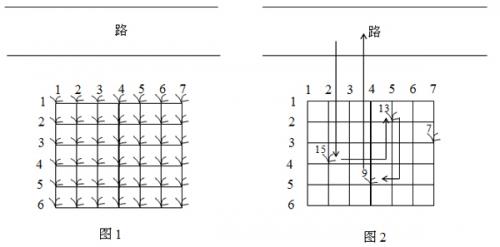
我们假定小Q在每个单位时间内，可以做下列四件事情中的一件：

1）从路边跳到最靠近路边（即第一行）的某棵花生植株；

2）从一棵植株跳到前后左右与之相邻的另一棵植株；

3）采摘一棵植株下的花生；

4）从最靠近路边（即第一行）的某棵花生植株跳回路边。



现在给定一块花生田的大小和花生的分布，请问在限定时间内，小Q最多可以采到多少个花生？注意可能只有部分植株下面长有花生，假设这些植株下的花生个数各不相同。

例如在图2所示的花生田里，只有位于(2,5),(3,7),(4,2),(5,4)的植株下长有花生，个数分别为13,7,15,9。沿着图示的路线，小Q在21个单位时间内，最多可以采到37个花生。

要求：采用贪心算法[[1]](#footnote-1)解决该问题。

|  |  |
| --- | --- |
| 输入数据文件“peanut.txt” | 输出 |
| 6 7 21 //6行7列21个单位时间  0 0 0 0 0 0 0  0 0 0 0 13 0 0  0 0 0 0 0 0 7  0 15 0 0 0 0 0  0 0 0 9 0 0 0  0 0 0 0 0 0 0 | 6 7 21  0 0 0 0 0 0 0  0 0 0 0 13 0 0  0 0 0 0 0 0 7  0 15 0 0 0 0 0  0 0 0 9 0 0 0  0 0 0 0 0 0 0  最多采到了37个花生 |
| 若输入的是：6 7 20 | 输出：最多采到了28个花生 |

## 2．舞会配对问题

**题A**  在化妆舞会上，男士们和女士们各自排成一队从不同的方向进入舞池。跳舞开始时，依次从男队和女队的队头上各出一人配成舞伴。若两队初始人数不相同，则较长的那一队中未配对者等待下一轮舞曲。试编写算法模拟上述舞伴配对问题，输出前三轮配对的男士和女士。**（**文件取名为**dancerA.c**）

|  |  |
| --- | --- |
| 输入数据文件“dancerA.txt” | 输出 |
| 12 //12人参加舞会  George Thompson M //M男士，F女士  Jane Adrews F  Sandra Williams F  Bill Brooks M  Bob Carlson M  Shirley Granley F  Louise Sanderson F  Dave Evans M  Harold Brown M  Roberta Edwards F  Dan Gromley M  John Gaston M | 第1轮舞伴：  Jane Adrews George Thompson  Sandra Williams Bill Brooks  Shirley Granley Bob Carlson  Louise Sanderson Dave Evans  Roberta Edwards Harold Brown  第2轮舞伴：  Jane Adrews Dan Gromley  Sandra Williams John Gaston  Shirley Granley George Thompson  Louise Sanderson Bill Brooks  Roberta Edwards Bob Carlson  第3轮舞伴：  Jane Adrews Dave Evans  Sandra Williams Harold Brown  Shirley Granley Dan Gromley  Louise Sanderson John Gaston  Roberta Edwards George Thompson |

**题B**  假设参加舞会的男士女士人数不一样，晚上共有10首舞曲，M先生想知道，哪几轮能跟F女士作舞伴。（文件取名为**dancerB.c**）

|  |  |
| --- | --- |
| 输入数据文件“dancerB.txt” | 输出 |
| 12 //12人参加舞会  Louise Sanderson //F女士  John Gaston //M先生  George Thompson M  Jane Adrews F  ︙ 同A  也可使用“dancerA.txt”，F女士和M先生从键盘输入或直接初始化。 | Louise Sanderson女士 和 John Gaston先生共舞  第3轮：Louise Sanderson John Gaston  第10轮：Louise Sanderson John Gaston  共2轮可以做舞伴。 |

## 3．会议安排问题

**题A** 某公司的会议日益增多，以至于全公司唯一的会议室不够用了。现在给出这段时期的会议时间表，要求适当删除一些会议，使得剩余的会议在时间上互不冲突，要求删除的会议最少。（文件取名为**meetA.c**）

|  |  |
| --- | --- |
| 输入数据文件“meetA.txt” | 输出 |
| 10 //有10个会议  800 830 //分别表开始时间和结束时间  715 840  900 930  830 1240  1150 1200  900 1210  1300 1400  1400 1600  1350 1500  1630 1800 | 会议安排如下:  开始时间 结束时间  800 830  900 930  1150 1200  1300 1400  1400 1600  1630 1800  取消会议4个 |

**题B** 如果有些会议是一定要安排的，不能被取消的，请重新设计会议安排问题。（文件取名为**meetB.c**）

|  |  |
| --- | --- |
| 输入数据文件“meetB.txt” | 输出 |
| 10 2 //10个会议，其中2个是不能取消的  830 1240  1300 1400  800 830  715 840  900 930  830 1240  1150 1200  900 1210  1300 1400  1330 1600  1450 1630  1600 1800 | 会议安排如下:  开始时间 结束时间  800 830  830 1240  1300 1400  1450 1630  取消会议6个 |

## 4．全排列问题

输出1，2，3，4四个数的所有排列方法及排列总数。拓展：输出n(n<=10)个**不同**的字符的所有排列方法及排列总数。（文件取名为**permutation.c**）

要求：采用分治法[[2]](#footnote-2)解决该问题。

|  |  |
| --- | --- |
| 输入测试样例1 | 样例1输出 |
| 以字符串的形式输入要全排列的字符：1234 | 1234 1243 1324 1342 1432 1423  2134 2143 2314 2341 2431 2413  3214 3241 3124 3142 3412 3421  4231 4213 4321 4312 4132 4123  全排列个数：24 |
| 输入测试样例2 | 样例2输出 |
| 以字符串的形式输入要全排列的字符：abc | abc acb bac bca cba cab  全排列个数：6 |

## 5．背包装载问题

有两只背包载重量分别为c1和c2，现有n本书要装入包中，每本书的重量用wi表示，并且满足∑wic1+c2，试找出合理的方法，将这n本书装入背包中。

要求：采用回溯法[[3]](#footnote-3)解决该问题。（文件取名为**bag.c**）

|  |  |
| --- | --- |
| 输入测试样例1 | 样例1输出 |
| 50 60 //表示c1,c2  4 //书的数量  20 15 20 30 //每本书的重量 | book 20 15 20 30  c1 1 0 0 1 //1表示装入  c2 0 1 1 0 |
| 输入测试样例2 | 样例2输出 |
| 50 60  4  10 30 35 35 | Unsolvable |

# 四、设计成果

1．源程序

提交每题的源程序代码。

2．课程设计报告(要求电子稿)

（1）需求分析

简述课程设计要解决的主要问题。在该部分中叙述，主要模块的功能要求。

（2）概要设计

确定算法使用的数据结构，对算法进行简单描述。

（3）详细设计

设计各算法实现的源程序，要求结构清晰，重点功能部分要加上注释。

（4）调试分析

使用各种数据进行程序测试，以保证程序的稳定性，并进行算法的空间复杂度和时间复杂度分析，每个模块设计和调试时存在问题的思考，算法的改进设想。

（5）总结

包括在课程设计过程中遇到的问题、解决问题过程中的思考和解决的方法、程序调试能力的思考、获得的经验，以及在课程设计过程中对《数据结构》课程的认识等内容。

# 五、成绩构成及评分标准

1．态度分（15分）：体现学习态度及遵守纪律情况。

1. 个人与团队表现分（15分）：体现独立工作能力及创新精神。

3．成果分（60分）：体现专业理论知识掌握程度、成果与答辩。

（1）功能设计（50分）：由老师根据学生的演示结果评定。

（2）答辩分（10分）：从学生现场回答问题的逻辑性进行评分。

4．设计报告（10分）：从规范要求上评定。

**数据结构与算法课程设计评分标准**

| **项**  **目** | **分值** | **优秀**  **（100≥X≥90）** | **良好**  **（90>X≥80）** | **中等**  **（80>X≥70）** | **及格**  **（70>X≥60）** | **不及格**  **（X<60）** |
| --- | --- | --- | --- | --- | --- | --- |
| **专业理论知识掌握程度** | **10** | 综合运用知识的能力和动手能力强 | 综合运用知识的能力和动手能力较强 | 有一定的动手能力，能够在一定程度上综合运用所学知识，但有所欠缺 | 综合运用所学知识的能力及实践动手能力较差 | 综合运用所学知识的能力及实践动手能力差 |
| **独立工作能力及创新精神** | **15** | 独立工作能力强，有独到见解，创新意识和创新能力高，并具有良好的团队协作精神 | 有一定的独立工作能力，并具有较好的团队协作精神 | 独立工作能力较差，有一定的团队协作精神 | 独立工作能力差 | 课程设计过程中有作弊行为，如抄袭他人成果、盗用他人数据等 |
| **学习态度及遵守纪律** | **15** | 态度认真，按设计任务书要求圆满完成规定任务；无迟到、早退和旷课 | 态度认真，按设计任务书要求完成规定任务；无迟到、早退和旷课 | 态度认真，按设计任务书要求完成规定任务；无迟到、早退和旷课 | 态度一般，在指导教师帮助下按期完成规定任务；旷课不超过全部设计时间三分之一及以上 | 态度不认真，未能按期完成规定任务；旷课超过全部设计时间三分之一及以上 |
| **成果与答辩** | **50** | 设计方案合理，计算、分析正确，设计质量高；答辩过程中思路清晰，论点正确，对题目理解深入，回答问题正确 | 设计方案合理，计算、分析正确，设计质量较高；答辩过程中思路清晰，论点基本正确，对题目理解较深入，主要问题回答正确 | 设计方案基本合理，计算、分析基本正确，设计质量一般；答辩过程中思路比较清晰，分析不够深入，主要问题回答基本正确 | 设计方案基本合理，计算、分析有错误，设计质量一般；答辩过程中分析问题较为肤浅，主要问题经启发后能回答 | 设计方案不合理，计算、分析有重大错误，设计质量差；答辩过程中无法回答问题 |
| **规范**  **要求** | **10** | 设计说明书条理清晰、论述充分、文字通顺、图纸图表规范、符合设计说明书文本格式要求 | 设计说明书条理清晰、论述正确、文字通顺、图纸图表较为规范、符合设计说明书文本格式要求 | 设计说明书条理基本清晰、论述基本正确、文字通顺、图纸图表基本规范、符合设计说明书文本格式要求 | 设计说明书条理不够清晰、论述不够充分、没有原则性错误、文字基本通顺、图纸图表不够规范、符合设计说明书文本格式要求 | 课程设计有原则性错误 |

1. 贪心算法（又称贪婪算法）是指在对[问题求解](https://baike.baidu.com/item/%E9%97%AE%E9%A2%98%E6%B1%82%E8%A7%A3/6693186)时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，它所做出的是在某种意义上的局部[最优解](https://baike.baidu.com/item/%E6%9C%80%E4%BC%98%E8%A7%A3/5208902)。我们可以通过各种方式找到当前的最优解，将最优解利用过后，将其清除，再去找下一个最优解。 [↑](#footnote-ref-1)
2. 分治法，字面上的解释是“分而治之”，就是把一个复杂的问题分成两个或更多的相同或相似的子问题，再把子问题分成更小的子问题……直到最后子问题可以简单的直接求解，原问题的解即子问题的解的合并。 [↑](#footnote-ref-2)
3. 回溯法（探索与回溯法）是一种选优搜索法，又称为试探法，按选优条件向前搜索，以达到目标。但当探索到某一步时，发现原先选择并不优或达不到目标，就退回一步重新选择，这种走不通就退回再走的技术为回溯法，而满足回溯[条件](https://baike.baidu.com/item/%E6%9D%A1%E4%BB%B6/1783021)的某个[状态](https://baike.baidu.com/item/%E7%8A%B6%E6%80%81/33204)的点称为“回溯点”。 [↑](#footnote-ref-3)